****



**PLC Applications**

Module 2: Sequential circuits and Step sequence





PREPARED BY

**Academic Services Unit**

December 2011

© Applied Technology High Schools, 2011

Module 2: Sequential circuits and Step sequence

**Module Objectives**

Upon successful completion of this module, students will be able to:

* Analyze conditional control tasks and develop proper solutions for such tasks.
* Understand and utilize relay-based safety circuit and use it with LOGO! Controller.
* Apply sequential control routines to malty-stage systems.
* Develop step sequence control routines using “Pulse relay” programming block.

**Module Contents:**

|  |  |  |
| --- | --- | --- |
|  | **Topic** | **Page No.** |
| 2.1 | Introduction | 3 |
| 2.2 | Conditional control | 4 |
| 2.3 | Sequential control | 6 |
| 2.4 | Pulse relay programming block | 14 |
| 2.5 | Practical tasks | 15 |
| 2.6 | Module exercises | 24 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| * 1. **Introduction**   In many applications control tasks are done in sequence. As an example: the mixing machine shown in figure 2.1 contains three units: feeder, mixer, and water circulator. It is very important to start the water circulator and the feeder units before starting the mixer unit. The reason for that is to avoid overloading the mixer unit.   |  | | --- | | untitled1.bmp | | **Figure 2.1 Mixing machine** |  |  |  | | --- | --- | | Another example is shown in figure 2.2; a pedestrian signal includes a countdown timer, which shows pedestrians how much time remains to cross the intersection. The pedestrian signal must stay red until the car signal goes red. | photo.JPG | |  | **Figure 2.2 Pedestrian signal** |   The question is: How sequential tasks can be done?  **2.2 Conditional control**  A programming instruction that tells the controller to execute part of a program only if conditions are true is called **conditional control routine**.  In the ladder diagram shown in figure 2.3, I1 a normally open pushbutton and I2 a normally closed pushbutton enables and disables Q1 respectively; while I3 a normally open pushbutton and I4 a normally closed pushbuttons enables and disables Q2 respectively.  Due to that the normally open contact of Q1 is connected to the circuit of Q2 in series, Q1 becomes an **AND condition** for Q2.  Therefore, only when Q1 is enabled can Q2 be enabled.   |  | | --- | | test.bmp | | **Figure 2.3 Conditional control routine** |   If the control task contains more than one condition, all conditions are to be added in series in the same rung.  In the previous ladder diagram switching OFF Q1 will immediately terminate Q2. In some applications a condition is required only to start an action then the action can be stopped alone. In this case a latch is to be used across the condition.   |  | | --- | | photo.JPG | | **Figure 2.4 Latching conditional control routine** |   **Lab activity 1 page: 15**  **2.3 Sequential control**  When tasks involved in a control system are to be done in a certain order control is said to be sequential control, this is very common in industrial applications, the following example explains the idea.  A grain store is to be controlled by LOGO! The grain is tipped into a pit and transported from there into a cyclone by means of an elevator (conveyor belt with scoops). The cyclone separates the corn from the chaff which is blown away. The heavier corn drops downwards and is transported by a conveyor worm into a silo.   |  | | --- | |  | | **Figure 2.5 Automatic grain store** |   To ensure that neither the cyclone nor the conveyor worm is overloaded, the system must be switched on as follows:  Conveyor worm 🡪 Cyclone 🡪 Elevator   |  |  | | --- | --- | | The following conditions apply:  • Switching-off must be carried out in the reverse order.  • Operation is only carried out using the control panel shown in figure 2.6.  • The emergency stop button can be used any time to stop the overall system. |  | | **Figure 2.6 Control panel** | |   Note : All green PBs are N.O and all red PBs are N.C  A safety circuit with EMERGENCY STOP must be provided separately.  **Safety circuit**  If the safety of a person or system is endangered, it must be possible to stop the plant safely and immediately. The following requirements apply:  • EMERGENCY STOP must have priority over all other operations  • The state resulting in the danger must be stopped without a power supply.  • Resetting of the EMERGENCY STOP must not result in switching-on of the plant again.  • EMERGENCY STOP must be carried out using electromechanical devices.  • EMERGENCY STOP is always carried out using an NC contact.   |  |  | | --- | --- | | The control circuit shown in figure 2.7 is a common safety circuit used in many industrial applications; in this circuit a normally open pushbutton (N.O PB) provides a current path when it is pressed, which will energize the relay coil. After releasing the ON PB, the relay coil stays energized because of the current path provided by the relay contact.  A normally open contact taken from the relay is connected in series with the output for safety.  This circuit can be used in our grain store example as a safety circuit. | subbing.bmp | | **Figure 2.7 Safety circuit** | |   **Lab activity 2 page: 19**  Now back to our sequential control example “the grain store”. A carefully produced list of connection names helps avoid errors and facilitates updating and maintenance of the controlled system.  The individual objects are then programmed and simulated. The individual objects are then combined step-by-step into a total system. The total system should be tested again before the software is used on the machine.  Table 2.1 shows the assignment list for the I/O of the grain store example:   |  |  |  | | --- | --- | --- | | **Inputs** | | | | Input | Name | Address | | Start conveyor worm | S1 | I1 | | Stop conveyor worm | S2 | I2 | | Start cyclone | S3 | I3 | | Stop cyclone | S4 | I4 | | Start elevator | S5 | I5 | | Stop elevator | S6 | I6 | | Switch ON PB | S7 | Safety circuit | | Emergency stop | S8 | Safety circuit | | Relay coil | K1 | I9 | | **Outputs** | | | | Output | Name | Address | | Conveyor worm | M1 | Q1 | | Cyclone | M2 | Q2 | | Elevator | M3 | Q3 | | Conveyor lamp | P1 | Q4 | | Cyclone lamp | P2 | Q5 | | Elevator lamp | P3 | Q6 |   **Table 2.1 I/O assignment list for grain store example**  Figure 2.8 shows a possible wiring diagram for the grain store example.   |  | | --- | | photo.JPG | | **Figure 2.8 Wiring diagram - grain store.** |   In this diagram three fuses are used for protection, F1 is used to protect the LOGO! PLC while F2 is used to protect the three motors (conveyor worm, cyclone and elevator) and the last fuse F3 is used to protect the three lamps. An expansion module is used to increase the number of digital output.  This wiring diagram contains three colors of wires:   1. Black: wires connecting inputs. 2. Green: wires controlling motors. 3. Blue: wires connecting lamps. 4. Red: emergency stop circuit.   It is clear from this wiring diagram that the emergency circuit has the priority over the LOGO! Controller. For safety reasons switching OFF is done using electromechanical device (Relay).  Now let us create a program for this control system using LOGO! Soft comfort software, first of all a latch is needed to start and stop each stage, figure 2.9 shows the first programming step   |  | | --- | | untitled1.bmp | | **Figure 2.9 First programming step – grain store** |   In the FBD shown figure 2.9 neither the main ON pushbutton nor the emergency pushbutton is used. In fact No need to include the main ON pushbutton in our FBD since the main ON PB is used to switch ON the relay of the safety circuit, and a normally open contact from this relay is connected in series with each motor, this will ensure that motors run only if the relay is ON.  According to the safety standards; resetting of the emergency stop must not result in switching-on of the plant again. That means pressing the emergency stop must also reset all latches, so we need to modify our FBD by including the emergency stop.  Figure 2.10 shows the modified version of the FBD with emergency stop.   |  | | --- | | untitled1.bmp | | **Figure 2.10 Grain store – with emergency stop** |   Still our three stages are not in sequence. Any stage can start at any time. But the system requires that Q2 can only be switched on if Q1 is already switched on and Q3 cannot be switched on before Q2.   |  |  | | --- | --- | | To create a **sequential-start** circuit, the output of the previous stage should be used as a **SET** input for the current stage as in figure 2.11 | untitled1.bmp | | **Figure 2.11 Sequential start** | |   To start the grain store stages sequentially, the output of the first stage becomes a SET input for the second stage, and the output of the second stage becomes a SET input for the third stage, the new FBD is shown in figure 2.12. This is similar to what we did in conditional control.   |  | | --- | | untitled1.bmp | | **Figure 2.12 Grain store – sequential start** |  |  |  | | --- | --- | | To create a **sequential-stop** circuit, the output of the next stage should be used as a **RESET** input for the current stage as in figure 2.13 | untitled1.bmp | | **Figure 2.13 Sequential stop** | |   To stop the grain store stages sequentially, the output of the third stage becomes a RESET input for the second stage, and the output of the second stage becomes a RESET input for the first stage, the new FBD is shown in figure 2.14.   |  | | --- | | untitled.bmp | | **Figure 2.14 Grain store** |   **Lab activity 3 page: 22**  **2.4 Pulse relay programming block**   |  |  | | --- | --- | | Pulse relay programming block has three inputs; like in the normal RS latch block input S is used to set the output Q to logic 1 and input R is used to set the output Q to logic 0.  Unlike the normal latch programming block in Pulse relay programming block there is a third input that is Trg. Input Trg is used to toggle the status of the output Q. | untitled.JPG | | **Figure 2.15 Pulse relay block** | |   High signal at Trg changes the output from ON to OFF or from OFF to ON when both S and R are 0, however input Trg does not influence the output when S = 1 or R = 1.  **Step sequence** is a sequential control circuit in which one step is done every time the input is triggered.  **Startup flag M8**  The M8 flag is set in the first cycle of the user program and it is reset after the first program execution cycle.  **Lab activity 4 page: 23** |

|  |
| --- |
| **2.5 Practical tasks**  **Lab activity 1 : Conditional control** |
| **Objective:** Apply conditional control routines |
| A conveyor belt goes ON and OFF using a switch. The Edutrainer table moves in the forward direction if the green pushbutton is kept pressed and it moves in the backward direction if the white pushbutton is kept pressed. Table moves **only** if the conveyor belt is ON.  Note: software-based interlock must be used to protect the motor table.  **Use The Edutrainer prototype production line to implement this control task.**   1. Create an I/O assignment list.  |  |  | | --- | --- | | **Inputs** | | | Input | Address | |  |  | |  |  | |  |  | | **Outputs** | | | Output | Address | |  |  | |  |  | |  |  |  1. What is the condition in this control task? 2. Analyze the system requirements and write the Boolean expression for each output.  |  |  |  | | --- | --- | --- | | **System requirements** | | | | Output | Requirements | Boolean expression | |  |  |  | |  |  |  | |  |  |  |  1. Draw the Ladder diagram for this control task.  |  | | --- | |  |  1. Use the LOGO! Soft comfort software to solve this task.   While solving this control task you are required to produce a connection table and to describe the task in the properties window.   1. Run and test the program. then fill in the table provided below by writing the status of the Edutrainer table:  * Moving forward * Moving backward * NOT moving  |  |  |  |  | | --- | --- | --- | --- | | **Conveyor belt is OFF** | | | | |  | Only green PB is kept pressed | Only white PB is kept pressed | Both green and white PBs are pressed | | Table status |  |  |  | | **Conveyor belt is ON** | | | | | Table status | Only green PB is kept pressed | Only white PB is kept pressed | Both green and white PBs are pressed | | Table status |  |  |  |  1. In the previous control task modify the ladder diagram so that the Edutrainer table can be switched OFF independently after switching OFF the conveyor belt. |

|  |
| --- |
|  |

|  |
| --- |
| **Lab activity 2: Safety circuit** |
| **Objective:** Apply safety circuit concept using relay   |  |  | | --- | --- | | Relay is an electrically operated switch. Current flowing through the coil of the relay creates a magnetic field which attracts a lever and changes the switch contacts.  Your teacher will provide you by a relay as the one shown in the figure. | IMG_0509.jpg | |  | **Figure 2.16 Relay** |  1. Fill in the table provided below by writing the assigned numbers:  |  |  | | --- | --- | |  | assigned Numbers | | Normally open contacts |  | | Normally closed contacts |  | | Common contacts |  | | Coil terminals |  |  1. How many normally open contacts and normally closed contacts are provided by this relay? 2. Contact 7 can be used with contact (3 , 11) as a normally ( open , close) contact. **(Circle the correct answer)** 3. Connect the circuit as shown in figure 2.17  |  |  | | --- | --- | | IMG_0511.jpg | untitled.bmp | | **Figure 2.17 Safety circuit without load** | |  1. Press the green pushbutton and write your observation. 2. Press the red pushbutton and write your observation. 3. Modify the previous circuit by connecting a lamp as shown in figure 2.18  |  |  | | --- | --- | | untitled.bmp | untitled1.bmp | | **Figure 2.18 Safety circuit with load** | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Write a simple PLC program so that the lamp in the previous circuit goes ON when the green PB at the Edutrainer is pressed and it stays ON until the red PB at the Edutrainer is pressed.  Draw your program in the space provided below.   |  | | --- | |  |  1. What is the use of the relay circuit in the previous task? 2. Turn OFF the lamp using the emergency PB in the safety circuit. And then turn the safety circuit ON, what have you noticed? Does this comply with the safety standards? 3. What are the modifications should be done so that the previous safety circuit complies with the safety standards.(draw the modifications)  |  |  | | --- | --- | |  |  | |

|  |
| --- |
| **Lab activity 3 : Sequential control** |
| **Objective:** Apply sequential control routines |
| Use the LOGO! Soft comfort software to create and simulate the FBD shown in figure 2.14 for the grain store example, and then answer the following questions.   1. Try to turn the elevator (Q3) ON while the conveyor worm (Q1) is OFF, Is that possible? 2. Turn ON all the stages sequentially, and then try to turn OFF the first stage, is that possible? 3. Use “Message texts” programming block to show which stage is ON.  |  | | --- | |  | |
| **Lab activity 4 : Step sequence** |
| **Objective:** Use step sequence control routines |
| Use the LOGO! Soft comfort software and Edutrainer kit to create and test the subroutines shown below and write your comments.   |  |  |  | | --- | --- | --- | | Subroutine | Actions | Comments | | sol.bmp | Press and release I1 several times. |  | | sol.bmp | Press and release I1 several times. |  | | sol.bmp | Press and release I1 several times. |  | |
| **2.6 Module Exercises** | |
| 1. A system has four stages is to be controlled sequentially using LOGO! Each stage has 2 pushbuttons one is N.O PB used to turn the stage ON the other is N.C PB used to turn the sage OFF.   Stages should start as follows: Q1🡪Q2🡪Q3🡪Q4,  stages should stop as follows: Q4🡪Q3🡪Q2🡪Q1,   1. Draw the FBD for this system without any emergency STOP. 2. Add an emergency stop to your FBD in the previous part. 3. Use the LOGO! Soft comfort software along with the Edutrainer kit to develop an FBD that can do the following steps (only one step should be done every time the green PB is pressed)   First step: Green light goes ON.  Second step: White light goes ON and green goes OFF.  Third step: Conveyor belt goes ON and white light goes OFF.  Fourth step: Conveyor belt goes OFF branching arm extends.  The same sequence repeats again. | |